A novel approach to reconstruction based saliency detection via convolutional neural network stacked with auto-encoder

Lin, Xinchen; Tang, Yang; Tianfield, Huaglory; Qian, Feng; Zhong, Weimin

Published in:
Neurocomputing

DOI:
10.1016/j.neucom.2019.01.041

Publication date:
2019

Document Version
Author accepted manuscript

A Novel Approach to Reconstruction based Saliency Detection via Convolutional Neural Network Stacked with Auto-encoder

Xinchen Lin[1], Yang Tang[1], Huaglory Tianfield[1], Feng Qian[1,**], Weimin Zhong[1,**]

Abstract

Visual saliency detection, toward the simulation of human visual system (HVS), has drawn much attention in recent decades. Reconstruction based saliency detection models are established for saliency detection, which predict unexpected regions via linear combination or auto-encoder network. However, these models are ineffective in dealing with images due to the loss of spatial information caused by the conversion from images to vectors. In this paper, a novel approach is proposed to solve this problem. The core is a deep reconstruction model, i.e., convolutional neural network for reconstruction stacked with auto-encoder (CNNR). On the one hand, the use of CNN is able to directly take two-dimensional data as input instead of having to convert the matrix to a series of vectors as in conventional reconstruction based saliency detection methods. On the other hand, the training process of CNN is augmented with the initialization obtained by an unsupervised learning process of convolutional auto-encoder (CAE). By this way, our CNNR model can be trained on limited labeled data, with the weights of the CNN being meaningfully initialized by CAE instead of random initialization. Performance evaluations are conducted through comprehensive experiments on four benchmark datasets and the comparisons with eight state-of-the-art saliency detection models show that our proposed deep reconstruction

*Corresponding author
**Corresponding author
Email addresses: linxinchenupup@163.com (Xinchen Lin), fqian@ecust.edu.cn (Feng Qian), wmzhong@ecust.edu.cn (Weimin Zhong)

model outperforms most of the eight state-of-the-art saliency detection models.
Keywords: convolutional neural network, auto-encoder, deep learning, reconstruction, saliency detection

1. Introduction

Human visual system (HVS) has a remarkable ability for handling complex scene, which can highlight salient objects in a complex scene and guide human to focus on them in a short time [1, 2]. Saliency detection is a concept inspired by HVS to process huge data rapidly, which has been widely applied in fields such as image/video compression, object detection, semantic segmentation, etc [3].

In HVS, processing of input data involves two complementary mechanisms: i.e., bottom-up and up-down. Bottom-up mechanism obtains salient regions based on low-level features such as color, texture, orientation, intensity and so on. Top-down mechanism requires high-level apriori knowledge to build models [4, 5]. As bottom-up mechanism requires no high-level information, it may be implemented more easily than up-down mechanism. However, as bottom-up mechanism is data driven, it may be less effective in some tasks, e.g., pedestrian detection and emotion recognition, which often suffer from occlusions and noises.

In HVS, another ability is to suppress background regions and highlight saliency regions. Reconstruction based saliency detection is to emulate this ability. In [6], Xia et al. proposed a reconstruction model to detect salient regions by computing the sparse reconstruction residual of central patches. The reconstruction process is realized by a linear combination of surrounding patches. However, this method does not consider the global rarity, and thus can not deal with the situation when local and non-local patches are close to each other. Later in [7], Xia et al. incorporated the reconstruction based method with the global rarity by introducing global competition to obtain the training data via a uniformly sampling strategy and then applied deep learning technique to train an auto-encoder network with the sampled data.

However, there is a major problem with the auto-encoder network. Since auto-encoder can not directly process images as two-dimensional inputs, an image has to be converted to a set of vectors before being inputted into the auto-encoder network. Such a conversion can cause a certain degree of loss of the spatial information in the image. Therefore, it's desired to find a way to avoid the conversion of images. Interestingly, it is noted that in the image classification, the convolutional neural network (CNN) is able to preserve the spatial information of the input image [8]. CNNs have been widely used in many fields, such as image classification [8], object detection [9] and sematic segmentation [10], etc., it is reasonable to anticipate that CNNs should work in the saliency detection as well.

In this paper, we propose a novel approach to reconstruction based saliency detection via convolutional neural network stacked with auto-encoder to avoid the conversion of images, we follow a bottom-up saliency detection framework that reconstructs local patches by a CNN. In our approach, we incorporate the global rarity into the framework of reconstruction based saliency detection using CNN. The core of our proposed approach is a deep reconstruction model, namely the convolutional neural network for saliency detection (CNNR), which infers the relationship between surrounding and central patches.

Compared with the existing reconstruction models, the novelty of our CNNR model is in two folds:

i) In our proposed CNNR model, feature extraction is performed by CNN during the training process adaptively. The CNN is capable of preserving the spatial information in an image without converting the matrix to vectors. As a result, our CNNR model is able to establish more exact representation of an image. To the best of our knowledge, this is the first time that the CNN is used in the reconstruction saliency detection.

ii) The training process of our CNNR model is augmented with the initialization obtained by an unsupervised learning process. An auto-encoder (AE) is leraned by unsupervised process in a convolutional way (thus called CAE)

and then the CAE stack is integrated to initialize the weights of CNN. The unsupervised process realized by CAE improves the performance where there is only limited labeled data for the training process of CNN.

The remainder of this paper is organized as follows: section II presents a review of the related work. Then we puts forward our deep reconstruction model in section III. Section IV conducts performance evaluations in comprehensive experiments and compares our proposed model with eight state-of-the-art saliency detection methods. Finally, section V draws up conclusions.

2. Literature Review

The literature review is presented in three strands, namely, center-surround contrast and global rarity, feature selection issues and saliency detection models.

2.1. Center-Surround Contrast and Global Rarity

Center-surround (C-S) contrast is a fundamental hypothesis in saliency detection, which treats a region as saliency when it is apparently contrasting to its surrounding regions. With the C-S contrast hypothesis, a visual saliency detection model was proposed by Itti et al. [11], through simulating the structure of the typical visual neurons. Saliency maps were computed via a Gaussian pyramid on various feature maps and then integrated to form the final map. However, the model of Itti et al. [11] only considered local patches and was unable to handle images with complex texture structure effectively. To address this problem, local and nonlocal C-S contrast strategies were employed to estimate the saliency region rather than merely relying upon local patches. In Borji and Itti [12], the feature difference between local and nonlocal patches was computed, and likewise in Seo and Milanfar [13] the matrix cosine difference between local and nonlocal feature matrixes was computed. Alternatively, all the surrounding regions were combined in a unified way to compare with the central region.

Another concept, global rarity is also employed to estimate the saliency region. In Hou and Zhang [14], the spectral residual of an image was computed in spectral domain by analyzing the log-spectrum of the image. In the saliency region detection method based on the regional contrast strategy [15], a full-resolution high-quality saliency map was established by assessing global rarity differences between distinct regions.

Recently, there have been attempts to integrate the C-S saliency and the global rarity saliency into one model. In [12], local and global patches were considered in different color spaces, the saliency map was computed by measuring the patch rarity in each color channel and all the maps were integrated to form the final map. Peng et al. [16] proposed a hybrid method which combined local and global saliency to detect image salient region.

2.2. Feature Selection in Saliency Detection

Feature selection is a fundamental work for saliency detection since different features enable distinct regions to be highlighted. In [11], a variety of features were integrated to obtain the saliency map. In [11, 17, 18], the traditional features were employed to obtain a saliency map. Since different features may highlight distinct saliency regions, some features were designed to pop out the saliency region in complex scenes, such as symmetry [19], gist [20] and local steering kernel [13]. In practice, it is hard to achieve satisfactory performance by just a few simple features for a complex scene. The more features are integrated into the model, the more accurate the saliency map obtained would be. There has been a trend to add more and more features like [21, 22]. However, with the number of features increasing, the computational complexity becomes much higher. Machine learning offers a good solution, which combines different features in an optimized way instead of just summating all the features with pre-defined weights. In computer vision, features obtained by machine learning appear more desirable than hand-crafted features. In [23], a set of basis functions were obtained by applying the independent component analysis (ICA) to a set of random patches, which were sampled from natural images. Similarly, a

dictionary of patches was learned from natural scenes in [12]. In [24], a feature transformation was learned, which regarded the saliency regions as sparse noises to obtain the saliency map.

2.3. Saliency Detection Models

Based on different features, saliency detection models can be built in different ways, typically by use of feature distribution, information theory and reconstruction strategy. Based on the difference between the distributions of features the saliency region can be estimated by using Kullback-Leibler Divergence (KLD) effectively. In Itti and Baldi [1] the difference between apriori and aposteriori distributions computed by KLD was estimated to detect the saliency region. KLD was also used to estimate the difference between central and surrounding patches in [25].

By use of information theory, new saliency detection models can be built. In Hou et al. [26] the information divergence was calculated by KLD in a surprise model. In [23] a bottom-up model of saliency detection was integrated with the information maximization theory. In particular, the saliency detection model is built from artificial neural circuit by considering the Shannon's self-information measure. In Gao and Vasconcelos [17] the saliency detection was viewed as a discriminant process. Through information-theoretic processing of the discriminant process, the explicit information between features and labels was discovered. In [27] a visual saliency detection model was built through estimating the activity of features by Incremental Coding Length (ICL). In Zhang et al. [18] a Bayesian network was built to locate the saliency region using the information in natural images.

Reconstruction strategies have been presented to predict saliency region. In Xia et al. [6] the reconstruction of the patch was realized by a linear combination, and salient regions were obtained by estimating the difference between the reconstructed patch and the original patch. In Ren et al. [28] a regularized feature reconstruction framework was presented to highlight salient regions for video. In Xia et al [7], deep learning was integrated into the saliency detection.

Specifically, an auto-encoder network was employed to extract the features and reconstruct the input image patches simultaneously.

Deep learning has great ability in feature extraction and representation, and has recently been employed to resolve the saliency detection problem. Following its successful application in computer vision, CNNs become the first choice to be used to build saliency models. In Li and Yu [29], a CNN was employed to extract multi-scale features to build a high quality visual saliency model. Similarly, in [30] a deep neural network framework was proposed, which combined low-level features with high-level features to capture the structured information and semantic context in complex scenes. Nevertheless, there are some drawbacks in the existing saliency detection models using CNN. Firstly, CNN is a supervised learning model, which means that a large amount of labeled data is required for the training process. However, obtaining labeled data is time consuming, which hinders the application of CNN in saliency detection. Secondly, even though the training data may be acquired from some data-sets, the generalization of the model is a problem in that when dealing with an image that is far from the training dataset, the CNN model may not work satisfactorily.

2.4. Discussion

In most of saliency detection methods, the C-S saliency and the global rarity saliency are computed separately. The methods which obtain the saliency map by combining the two strategies will achieve a better performance. A lot of hand-crafted features have been applied to compute the saliency map in different models. However, the features learned and selected by machine learning give a better prediction of salient regions in images. Comparatively, the reconstruction saliency shows some desirable characteristics. The reconstruction methods combining the C-S saliency with the global rarity saliency by the global sampling strategy achieve a better performance than other methods. In this paper, the feature learned by CNN will be used to reconstruct the saliency map, called CNNR model.

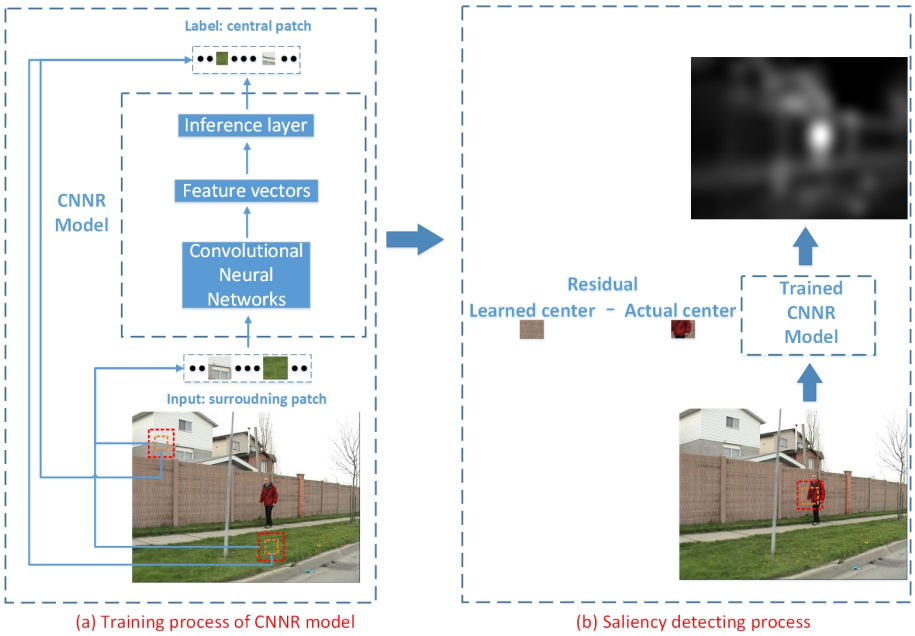

Figure 1: Deep reconstruction, i.e., convolutional neural network for reconstruction.

3. Deep Reconstruction

To address the above-mentioned problems in current reconstruction based saliency detection models, we propose a novel approach to reconstruction based saliency detection by using the CNN that is trained just by original image patches but not necessarily by any label information. Just learning from images themselves without collecting labeled data, our proposed approach turns out to be a more effective way to solving the saliency detection problem. What's more, the reconstruction strategy makes it a saliency model per image, which will avoid the performance degeneration when dealing with images those are far from the training dataset.

The core of our proposed saliency detection framework is the deep reconstruction using the CNN (called CNNR). As the CNN obtains an abstract representation of input images during the training process, the CNNR model is able

to reconstruct the central patches with the abstract representations obtained by the CNN. The final saliency map is obtained by computing the residual between the original central patches and the reconstructed patches obtained by the C-NNR model. Our proposed saliency detection framework can be illustrated in Fig. 1, which is composed of two parts, i.e., training of the CNNR model and saliency detection. In the training process of the CNNR model, a multi-layer convolutional neural network is at the bottom of the CNNR model. Feature vectors obtained by the CNN are inputted to the inference layer to learn a pattern to reconstruct central patches versus surrounding patches. The pattern learned from the sampled patches is used to detect the saliency for the same image. In the saliency detecting process, the same image is inputted to the CNNR model. Finally a saliency map is obtained by computing the residual between the reconstructed central patches and the original central patches.

3.1. Inference Structure of Convolutional Neural Network

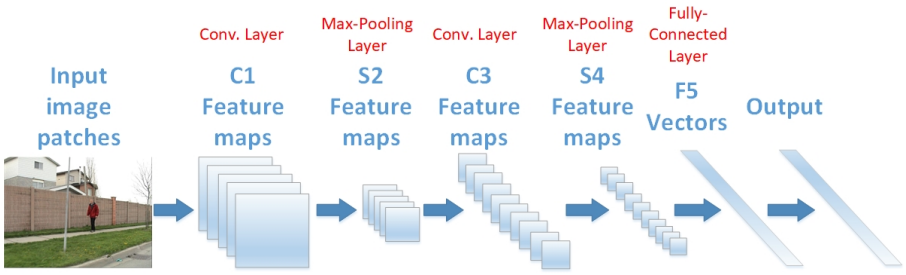

Figure 2: The inference structure of convolutional neural network in the proposed deep reconstruction model. The CNN has two convolutional layers: C1 and C3. Each convolutional layer is attached with a max-pooling layer: S2 and S4. There is a fully-connected layer F5 at the top of the S4.

Among various deep learning models, deep CNNs demonstrate particular ability in image processing. CNNs are able to handle an image as two-dimensional input directly, without necessarily converting the image matrix to a vector. Such a direct handling of an image as 2-D input will avoid the loss of spatial infor-

mation of the image during the conversion process. This is why we apply CNN to the deep reconstruction model for saliency detection.

The inference structure of the CNNR model is illustrated in Fig. 2. The training data in saliency detection are image patches, which are sampled from the original images. These patches are much smaller than the original images which are used as input in traditional applications of CNN. For this reason, the size and number of convolutional filters in our CNNR model are adjusted to an appropriate setting. At the top of CNN, a fully connected layer is appended to infer the relationship between the abstract representations of the surrounding patches and the central patches.

3.2. Training of the Inference Structure

After the inference structure has been built, the next step is to collect the training data. Firstly, a set of pixels are sampled from the original images randomly and uniformly. Then, the central patches and the surrounding patches which surround the sampled pixels make up the training data. The surrounding patches are regarded as the inputs to the inference structure and the central patches are as the labels. The process of collecting the central and the surrounding patches is the so-called global sampling strategy, which transforms the saliency detection problem to a sampling problem and inherently combines the C-S contrast theory with the global rarity. It should be pointed out that the global sampling strategy is based on the hypothesis that only a few areas of an image will make up the saliency regions, while most areas are background. When pixels are sampled from the original images randomly and uniformly, most of the corresponding patches are more likely to be background. In other words, the inference structure will learn a pattern which reconstructs central patches versus background regions.

To normalize the contrasts among different patches, first of all input images are transformed to the opponent color space by the method in [31] and normalized to [0, 1]. The channels are independent in the opponent color space. Then, images in various sizes are resized to the same scale appropriate to the CNNR

model.

After preprocessing of the image, sampled patches can be employed to train the network. However, there may be some similarities and relevance between the sampled patches. It is difficult to optimize the weights of CNN on limited data in supervised training process by using backpropagation. To improve the accuracy, a convolutional auto-encoder (CAE) is stacked to initialize the weights of the CNN in an unsupervised process.

CAE differs from the fully connected auto-encoder (AE) in that CAE directly takes two-dimensional matrix of an image as input, considering the spatial locality just as CNNs [32]. CAE adopts the idea of encoding and decoding in auto-encoder networks to the convolution neural networks. It encodes an image to different channels of feature maps, which are decoded to the original input image. In this way, CAE can be trained unsupervised and stacked to obtain a deep network. The weights of CAE stacked deep network can be used to initialize a CNN, which has the same architecture of the deep network. This initialization by the unsupervised pre-training of CAE will be better than the random initialization which is usually used in CNN and improve the accuracy of the CNNR model.

For a mono-channel input x, the latent representation of the k-th feature map can be obtained as below

$$y^k = \sigma(x * W^k + b^k) \tag{1}$$

where b^k is the bias for the k-th feature map, W is the weight which can be interpreted as the convolutional filter, σ is the activation function, and asterisk $*$ denotes the 2-dimension convolution operation. After the convolution operation, a set of feature maps y are obtained. Then the de-convolution operation reconstructs the input as below

$$z = \sigma(\sum_{k \in Y} y^k * \tilde{W}^k + c) \tag{2}$$

where c is the bias, Y is the set of feature maps obtained by (1), $\tilde{W}$ is the weight which can be interpreted as the de-convolutional filter. Since the

reconstruction should have the same size as the input, the convolution and de-convolution operations have to be set in different patterns. The convolution of a $p \times p$ matrix with a $q \times q$ matrix produces a $(p+q-1) \times (p+q-1)$ matrix, called full convolution. The de-convolution of a $p \times p$ matrix with a $q \times q$ matrix produces a $(p-q+1) \times (p-q+1)$ matrix, called valid convolution. The two operations are illustrated in Fig. 3. After the two operations, the reconstructed feature maps will have the same size as the input feature map. This is the basic principle of convolutional auto-encoder. Finally, weights and biases are optimized by minimizing the penalty functions below

$$P(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (x_i - z_i)^2 \tag{3}$$

$$\frac{\partial P(\theta)}{\partial W^{\mathrm{k}}} = x * \delta y^k + y^k * \delta z \tag{4}$$

where δy and δz denote the error term of hidden layer and output layer of CAE, which are computed to transmit from the output layer to input layer by back propagation algorithm. As the stochastic gradient descent (SGD) works well to optimize the weights in auto-encoder networks and deep neural networks, the weights W are updated by SGD in our CNNR model. Just as a fully connected AE which can be stacked to get a deep network, a CAE can also be stacked layer by layer to get a deep network. As the CAE learns an auto-encoder in a convolutional way, the CAE can also be stacked to initialize the CNN in a pre-training stage. After the pre-training process, the weights of CNN are optimized in the supervised training stage by backpropagation.

3.3. Saliency Detection By CNNR Model

After the training stage, a unified process is established to estimate the saliency. The process takes the surrounding regions as input and outputs the reconstructed central regions in the form of vectors. An original central patch with n channels is converted to a vector $c(x)$ by stacking all the columns of the patch in n channels. Finally, the saliency is estimated by computing the

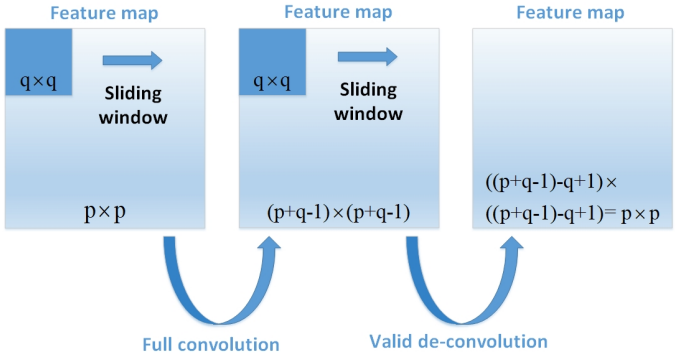

Figure 3: Full convolution operation and valid de-convolution operation.

residual between the predicted central regions and the original central regions as below

$$S(x) = \|f(s(x)) - c(x)\|_2 \tag{5}$$

where x denotes the sampled pixels from the image, $s(x)$ and $c(x)$ represent the vectors transformed from the surrounding patches and the central patches, respectively, which are corresponding to the pixels x. $f(\bullet)$ is the function learned by the CNNR model during the training process. The ι_2-norm is adopted to evaluate the difference between the vector of reconstructed and the vector of original regions. The saliency detection process by the CNNR model can be illustrated in Fig. 4. The saliency detection by CNNR model is divided into three parts: image preprocessing, training process and detecting process.

Our method is based on the center-surround contrast hypothesis that the saliency is where it is unexpected and different from the surroundings. A region is deemed unexpected if its reconstruction obtained by the CNNR model differs from the original region. To sum up, our proposed CNNR model has the following advantages. On the one hand, the CNNR model combines the reconstruction strategy with CNN, which builds the model just by the images themselves and detects the saliency region without any label information. Our CNNR model takes advantage of the powerful ability of CNN to abstract features from images directly without necessarily converting them into vectors. On the other hand,

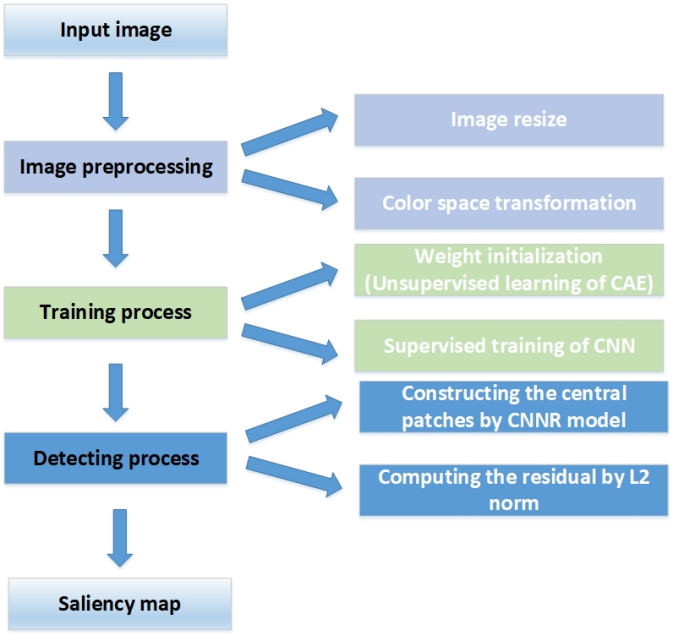

Figure 4: The flow-chart of saliency detection by the CNNR model.

the CNN is normally trained by supervised process. Our CNNR model, in order to be trained on limited labeled data, is initially augmented by the unsupervised learning process, which initializes weights of CNN meaningfully instead of random initialization.

4. Performance evaluation

In this section, comprehensive experiments are conducted to evaluate the proposed approach. The CNNR model is compared with eight state-of-the-art methods qualitatively and quantitatively. All the experiments are implemented on a computer with a 2.6GHz Intel i7-6700HQ CPU.

4.1. Datasets and evaluation metrics

Our CNNR model is tested on four well-known eye fixation datasets, namely Toronto, MIT, Kootstra and DUT-OMRON. The Toronto dataset is taken from Bruce and Tsotsos [23]. The fixation data are collected from 120 color images

Table 1: DATA-SETS

Dataset	Number of images	Average viewers	Resolution
MIT [23]	120	20	681*511
AIM [21]	1003	15	Various
Kootstra [33]	101	31	1024*768
DUT-OMRON [34]	5168	5	Various

with a constant resolution by 20 subjects. The MIT dataset is taken from Judd et al. [21], which is more abundant than the Toronto dataset. The MIT dataset consists of 1003 color images with various resolutions and fixation data from 15 subjects. The Kootstra dataset is taken from Kootstra et al. [33]. The fixation data of 101 images with various resolutions is obtained by 31 subjects. The DUT-OMRON dataset is from Yang et al [34], which consists of 5168 images with various resolutions. The fixation data of each image is collected by 5 subjects. The four datasets are outlined in Table 1.

Three evaluation metrics are adopted in this paper, namely Area under the curve (AUC), Shuffled AUC (sAUC) and Normalized Scanpath Saliency (NSS). The three metrics consider various measurements such as location, value and distribution.

AUC indicates the area under receiver operating characteristic (ROC) curve. The ROC curve is mostly applied to estimate the performance of classifiers according to different thresholds. Firstly, pixels in the saliency map are split into positive and negative samples through different thresholds. Compared with the fixation data, the true positive rate (TPR) and false positive rate (FPR) can be computed. Then the ROC curve is obtained by plotting the TPR against the FPR. The final AUC score is the average AUC score of all the images in the dataset.

In [18, 35], Zhang and Tatler argued that human will pay more attention to the center regions of an image. This is the so-called center-bias effect (CB) which has a strong influence upon the evaluation of original AUC score. In

light of the CB effect, Zhang and Tatler proposed an extension metric of AUC, called Shuffled AUC (sAUC). The main difference between the two metrics is the sampling strategy. sAUC extracts the negative samples from all fixations of the whole dataset rather than the current image.

Normalized scanpath saliency (NSS) was proposed by Peters et al. [36], which is used to estimate the degree of similarity between the saliency map and the human fixations. NSS estimated the relation between the fixation locations of human and the corresponding points on the saliency map. Firstly, the saliency map is normalized to have zero mean and unit standard deviation. Then, the saliency points are selected according to the fixation locations of human and the mean value of these points are computed. NSS is computed as below

$$NSS = \frac{1}{N}\sum_{i=1}^{N}\frac{S(x_f^i) - \mu_s}{\sigma_s} \tag{6}$$

where $S(x)$ represents the saliency map obtained by the CNNR model, x_f^i represents the selected saliency points according to the human fixation locations, μ_s is the mean and σ_s is the standard deviation of the saliency map, N is the number of the selected saliency points.

A high value of AUC and sAUC indicates a high classification ability of the model. A NSS value much greater than zero means a high correspondence between the saliency map and human fixation locations. On the contrary, a value much lower than zero suggests that an anti-correspondence between the predicted saliency points and human fixation locations. The value close to zero indicates no such relation.

4.2. Best model structure

In this section, comprehensive experiments are conducted to study the best structure of the CNNR model and the influence of different parameters in the global sampling stage. The structure of the CNNR model mainly depends on the CNN. The key parameters in the global sampling stage include the sampling number and size of sampled patches. For the sake of efficiency, the Toronto dataset is used to test the best structure and different parameters.

Table 2: STRUCTURES OF CONVOLUTIONAL FILTER

	Convolutional filter number of C1	Convolutional filter number of C3
Model 1 (5-10)	5	10
Model 2 (5-15)	5	15
Model 3 (10-20)	10	20

First of all, the learning parameters are set to constant. Since we have tested some settings of the training batch size and training epoch, we found it may may be insensitive to learning parameters for CAE and CNN. In the following experiments, the training batch size and training epoch are set to 100 and 10, respectively.

4.2.1. Structure of CNNR model and sampling number

The CNNR model is a CNN based reconstruction model for saliency detection, which is greatly influenced by the structure of CNN. The number of convolutional filters is a key factor of CNN, which has a great impact upon its performance. The structure of CNN in our approach is simpler than the models used in the traditional computer vision field. This is because in most of the traditional problems such as image classification, object detection and semantic segmentation, the input of CNN is a full resolution image. However, in our CNNR model, the input of CNN is sampled patches of the image, which are much smaller than the full resolution image. As a result, the number of convolutional filters in the CNNR model is smaller than the models in the traditional fields. Three different structures are tested for the number of convolutional filters. The settings of the three models are presented in Table 2.

Since the proposed CNNR model is trained by the sampled patches, the sampling number and the size of sampled patches also have an important effect on the performance. For each structure, 500-16000 patches are sampled to generate a set of datasets. The size of sampled patches is set to 15-7 (surrounding patch size - central patch size). The AUC scores versus sampling numbers and structures are plotted in Fig. 5. It can be seen that the proposed CNNR model

returns the best performance when the number of convolutional filters is set to 10-20 and the sampling number is 16000. In terms of accuracy and efficiency, Model 1 (5-10) is applied as the default setting in the comparative experiments with other methods.

According to Fig. 5, Model 3 (10-20) requires more sampled patches to return higher accuracy than the other two. This means that we need to sample more training patches to show the advantage of complex model. On the other hand, with the complexity of the CNNR model increasing, the performance only gains a little improvement. Therefore, it's reasonable to set the model to a relatively simple structure.

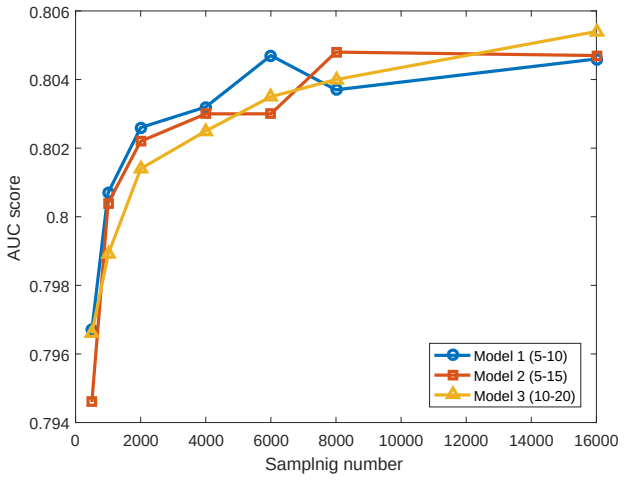

Figure 5: Impacts of different convolutional neural network structures (patch size: 15-7). Model 1, the number of convolutional filter is 5-10. Model 2, the number of convolutional filter is 5-15. Model 3, the number of convolutional filter is 10-20.

4.2.2. Size of sampled patches

The sampling size of the surrounding and central patches also affects the performance of CNNR model. Saliency and background regions may have totally varied sizes in different scenes. Different patch sizes will result in different

Table 3: SAMPLED PATCH VERSUS STRUCTURE OF CONVOLUTIONAL FILTER

	Surrounding patch size (S)	Central patch size (C)	Convolutional filter size of C1	Convolutional filter size of C3
Model 1 (15-7)	15	7	4	3
Model 2 (15-11)	15	11	4	3
Model 3 (21-7)	21	7	6	5
Model 4 (21-11)	21	11	6	5

patterns being learned by the network and the final saliency map will be affected. The size of sampled patches also has impact upon the size of convolutional filters in CNNR model. Once the former has changed, the latter will have to change accordingly. The patch size 15-7 is tested in the last section. In this section, three different settings for the size of sampled patches are tested, i.e., 15-11, 21-7 and 21-11. When the size of sampled patch is 15-7 or 15-11, the size of convolutional filters is set to 4 and 3, respectively. When the size of sampled patch size is 21-7 or 21-11, the size of convolutional filters is set to 6 and 5. The settings of the sizes of sampled patches and convolutional filters are shown in Table 3. In this comparative experiment, the number of convolutional filters is set to 5-10. The AUC scores versus size of sampled patch and sampling number are plotted in Fig. 6. It can be seen that the model returns the best performance when the sampled patch size is set to 15-11.

Based on the study of impacts on different parameters conducted above, we have the best settings for CNNR model. In terms of accuracy and efficiency, the structure of CNN should be set to 5-10, the sampling number and sampled patch size should be set to 8000 and 15-11, respectively.

Although the CNN architecture in this paper is relatively simple, we primarily demonstrate the applicability of CNN in reconstruction based saliency detection. With the increased complexity of images, the more complex model will meet requirements.

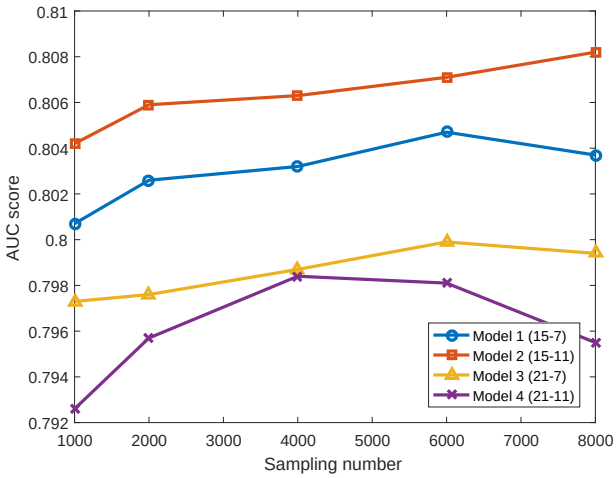

Figure 6: Experiments of different sampled patches size.

Table 4: Models FOR COMPARISONS

Saliency detection model	Description	Is center bias effect integrated?
Itti	difference between different features being computed	NO
AIM	information maximization theory	NO
SR	spectral residual in spectral domain being computed	NO
ICL	a dynamic visual attention model based on the rarity of features	NO
SUN	a Bayesian framework to locate the salient regions	NO
CA	context-aware model to detect salient regions	NO
GBVS	graph based visual saliency model	YES
AER	auto-encoder reconstruction saliency detection model	YES

4.3. Performance comparisons

To evaluate the performance, our CNNR model is compared with eight state-of-the-art methods, i.e., Itti's method (IT)[11], attention based on self-information (AIM) [23], spectral residual (SR) [14], incremental coding length (I-CL) [27], context-aware model (CA) [37], saliency using natural statistics (SUN) [18], graph based visual saliency (GBVS) [38] and auto-encoder reconstruction saliency (AER) [7]. These models are listed in Table 4.

Firstly, a qualitative comparison is conducted among saliency maps obtained by different methods. The saliency maps of several images are obtained by a

few representative saliency detection methods. However, the pixels highlighted in the saliency maps produced by different models will range in various scales. Some methods prefer to highlight many salient pixels, while others only highlight a few salient pixels. The method in [39] is employed to address this issue. The final results are shown in Fig. 7.

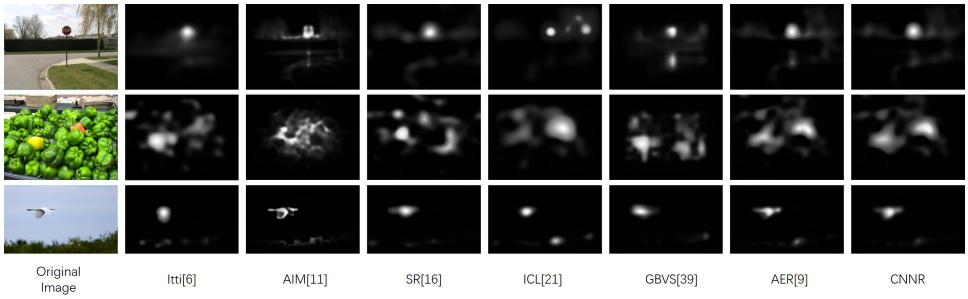

Figure 7: The saliency maps obtained by different methods.

Then the eight state-of-the-art methods are compared with our CNNR model quantitatively on the four datasets (Table 1). Most of these methods are not integrated with center-bias (CB) item, including the classic saliency detection methods in Itti [11] and AIM [23]. Some other methods like SR [14], ICL [27], SUN [18] and CA [37] also take no account of the CB effect. To make a fair comparison, we apply the CB item in our CNNR uniformly to these methods in the comparative experiments. Some methods have already integrated the CB item in themselves, such as GBVS [38], AER [7] and CNNR model. A smoothing technique is employed to saliency maps through a simple constant Gaussian kernel in CNNR model. Also for fair comparison, the same operation is applied to all the other methods that do not have this item. The performance (AUC, sAUC, NSS) of different methods on the four datasets are presented in Table 5, 6, 7 and 8, respectively. The ROC curves are plotted in Fig. 8 - 11, respectively.

It can be seen that the CNNR model outperforms most of the state-of-the-art methods on the four datasets. The CB item will enhance the AUC and NSS scores, but will reduce the sAUC scores of the models. Compared

Table 5: COMPARISON OF DIFFERENT METHODS ON DATA-SET #1 (best value is framed)

	AUC	sAUC	NSS
Itti	0.7947	0.6632	1.2027
AIM	0.8026	0.6753	0.9405
SR	0.7765	0.6909	1.2704
ICL	0.7997	0.6707	1.3363
SUN	0.7872	0.6662	1.1934
CA	0.8184	0.7035	1.5052
GBVS	0.8182	0.6398	1.4189
AER	0.8052	**0.7319**	1.3233
CNNR	**0.8196**	0.7185	**1.5802**

Table 6: QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON DATA SET #2 (best value is framed)

	AUC	sAUC	NSS
Itti	0.7682	0.6547	1.0603
AIM	0.7750	0.6750	0.9320
SR	0.7545	0.6720	1.0821
ICL	0.7796	0.6422	1.1617
SUN	0.7745	0.6855	1.1535
CA	0.8019	0.7008	1.3122
GBVS	**0.8084**	0.6572	1.2869
AER	0.7833	0.7090	1.1712
CNNR	0.8036	**0.7150**	**1.3648**

Table 7: QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON DATA SET #3 (best value is framed)

	AUC	sAUC	NSS
Itti	0.6592	0.6118	0.5760
AIM	0.6676	0.5975	0.5194
SR	0.6400	0.5689	0.4819
ICL	0.6679	0.5982	0.5878
SUN	0.6496	0.5930	0.5497
CA	0.6576	0.6203	0.5827
GBVS	0.6659	0.5986	0.5904
AER	0.6647	0.6320	0.6413
CNNR	**0.6692**	**0.6359**	**0.7043**

Table 8: QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON DATA SET #4 (best value is framed)

	AUC	sAUC	NSS
Itti	0.8272	0.7170	1.4618
AIM	0.8351	0.6982	0.9931
SR	0.7923	0.7073	1.4153
ICL	0.8139	0.6884	1.6102
SUN	0.8185	0.7015	0.3794
CA	0.8243	0.7429	1.4709
GBVS	**0.8530**	0.6938	1.6010
AER	0.6647	0.6320	0.6413
CNNR	0.8423	**0.7484**	**1.7283**

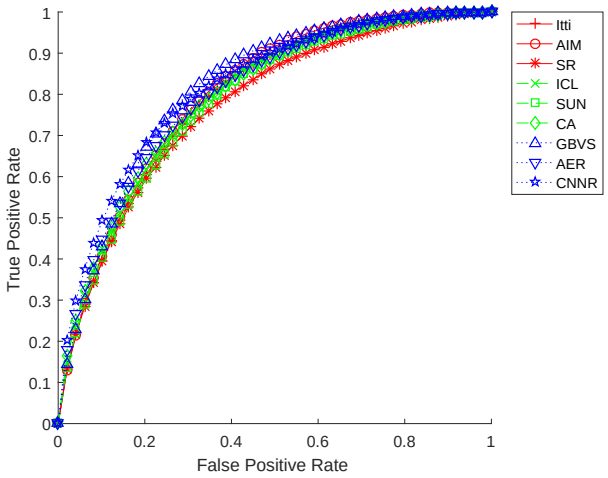

Figure 8: The ROC curves on data-set #1.

with AER, which is another reconstruction based saliency detection method, our proposed CNNR model shows another fantastic characteristic, that is, to achieve a given performance, our CNNR model is much faster, needs much less time. For example, for an image in dataset 1, the AER model takes 125.3 seconds to detect the image, whereas our CNNR model only takes 36.6 seconds for the same image.

5. Conclusion and future work

Our proposed method forms a saliency detection model, which reconstructs the saliency map using CNN. The differences between our proposed CNNR model and other models can be concluded as follows: firstly, the CNNR model based on the reconstruction strategy computes the C-S saliency and the global rarity saliency simultaneously, whereas other models like [11, 12, 13, 14, 15] obtain saliency map by considering just one of them. Secondly, the CNNR model extracts the feature by CNN instead of the hand-crafted features [11, 17, 18]. Thirdly, compared with other reconstruction strategies which use linear combination or auto-encoder network [6, 7], the CNNR model reconstructs the saliency

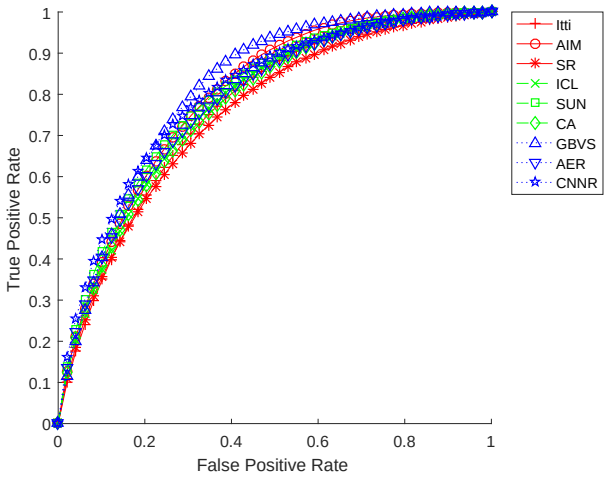

Figure 9: The ROC curves on data-set #2.

by the convolutional neural network, which preserves the spatial relationship in the images. It can thus be stated that the unsupervised learning of CAE and the deep learning of CNN constitute very promising solution to saliency detection.

In this paper, we have proposed a novel approach to reconstruction based saliency detection via convolutional neural network stacked with auto-encoder. The core is the deep reconstruction model called CNNR, which combines the deep learning technique of CNNs with convolutional auto-encoder for the reconstruction residual strategy. In this deep reconstruction model, CNNs are used to take an image directly as input without necessarily converting the image to a series of vectors, thus effectively preserving the spatial relationship of the image. In the pre-training stage of CNN, a convolutional auto-encoder (CAE) is stacked to initialize the weights of CNN. The effectiveness of the proposed CNNR model has been demonstrated through comprehensive experiments on four datasets. It has been showed that our CNNR model outperforms most of the state-of-the-art saliency detection methods. Overall, our work has demonstrated that the deep learning technique is a very promising solution to reconstruction based saliency detection. To the best of our knowledge, this is the first time that the CNN is

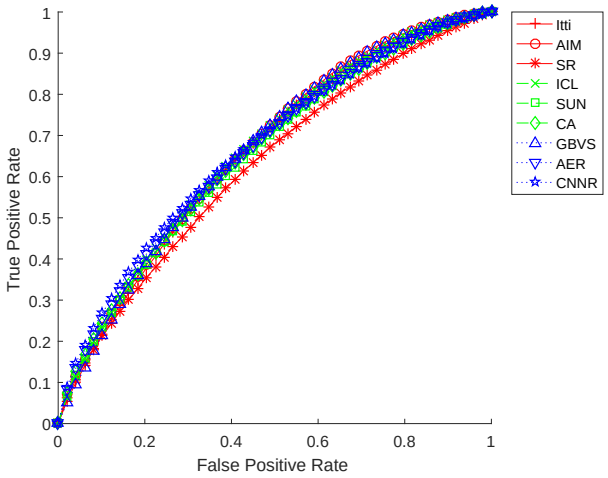

Figure 10: The ROC curves on data-set #3.

used in the reconstruction saliency detection.

Our proposed deep reconstruction model, CNNR, has the following strengths:

i) CNNR model is able to establish more exact representation of an image. On the one hand, feature extraction is performed by a CNN during the training process adaptively. In this way, our method will be able to fast achieve a better performance, not like the traditional ones which usually are burdened with searching for better features in dealing with complex scenes. On the other hand, compared with other reconstruction models, the CNN employed in our model is able to preserve the spatial information of images, which realizes a more exact feature representation.

ii) The training process of our CNNR model is augmented with the initialization obtained by an unsupervised learning process of convolutional auto-encoder. If the CNN training is undertaken through a normal supervised process, the number of sampled center-surround (C-S) patches would be far from enough to tune CNN's weights and biases. Instead, in our CNNR model, the unsupervised learning of an auto-encoder is first carried out in

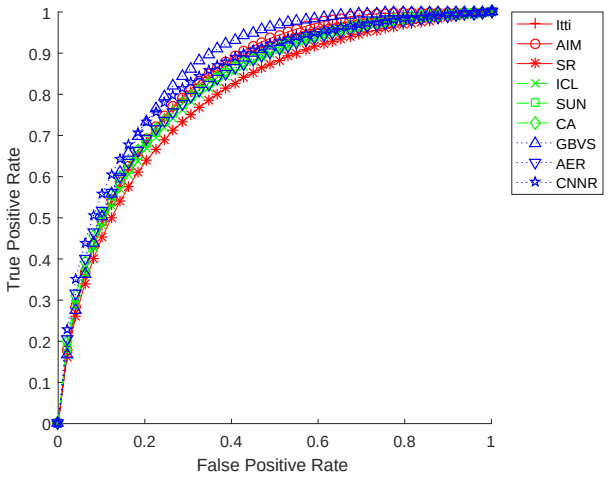

Figure 11: The ROC curves on data-set #4.

a convolutional way, thus called convolutional auto-encoder (CAE). Then, the learned CAE stack is integrated to initialize the CNN's weights. More meaningfully, by this way, our CNNR model can be trained on limited labeled data.

We see that a number of future works may be carried out to the CNNR model. Firstly, the size of sampled patches is fixed in this paper. This is because the input to CNN has to be a fixed-size image. As the sizes of saliency regions may vary from images to images, the fixed-size of sampled patches in the input may undermine the accuracy of model. Though different patch sizes are tested in this paper, the fixed-size settings may not be best suited for each image. A future work can be to introduce a new structure of the network to take varying size patches as input.

Secondly, our proposed CNNR model uses the global sampling strategy to generate the training dataset. This is based on the hypothesis that just a few salient regions should be sampled in the sampling stage. However, the CNNR model may lead to unexpected detection results when salient regions occupy most of an image.

Thirdly, as the CNN and CAE can be implemented easily by the Deep Learn Toolbox [40], the code of CNNR model is implemented by MATLAB. However, the speed is slower than implementation in C++. In the deep learning community, the caffe framework based on C++ is used to build deep CNNs which can be trained by GPU. We believe that an interesting future work would be to build much larger and deeper network for saliency detection.

References

[1] L. Itti, P. Baldi, Bayesian surprise attracts human attention, Vision research 49 (10) (2009) 1295–1306.

[2] J. M. Wolfe, T. S. Horowitz, What attributes guide the deployment of visual attention and how do they do it?, Nature reviews neuroscience 5 (6) (2004) 495–501.

[3] A. Toet, Computational versus psychophysical bottom-up image saliency: A comparative evaluation study, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (11) (2011) 2131–2146.

[4] S. K. Ungerleider, L. G, Mechanisms of visual attention in the human cortex, Annual review of neuroscience 23 (1) (2000) 315–341.

[5] A. Borji, D. N. Sihite, L. Itti, What/where to look next? modeling top-down visual attention in complex interactive environments, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (5) (2014) 523–538.

[6] C. Xia, F. Qi, G. Shi, P. Wang, Nonlocal center–surround reconstruction-based bottom-up saliency estimation, Pattern Recognition 48 (4) (2015) 1337–1348.

[7] C. Xia, F. Qi, G. Shi, Bottom–up visual saliency estimation with deep autoencoder-based sparse reconstruction, IEEE transactions on neural networks and learning systems 27 (6) (2016) 1227–1240.

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[10] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE transactions on pattern analysis and machine intelligence 39 (4) (2017) 640–651.

[11] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on pattern analysis and machine intelligence 20 (11) (1998) 1254–1259.

[12] A. Borji, L. Itti, Exploiting local and global patch rarities for saliency detection, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 478–485.

[13] H. J. Seo, P. Milanfar, Static and space-time visual saliency detection by self-resemblance, Journal of vision 9 (12) (2009) 15–15.

[14] X. Hou, L. Zhang, Saliency detection: A spectral residual approach, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[15] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, S.-M. Hu, Global contrast based salient region detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (3) (2015) 569–582.

[16] Y. Fang, Z. Fang, F. Yuan, Y. Yang, S. Yang, N. N. Xiong, Optimized multioperator image retargeting based on perceptual similarity measure, IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[17] D. Gao, N. Vasconcelos, Bottom-up saliency is a discriminant process, in: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–6.

[18] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, G. W. Cottrell, Sun: A bayesian framework for saliency using natural statistics, Journal of vision 8 (7) (2008) 32–32.

[19] G. Kootstra, A. Nederveen, B. De Boer, Paying attention to symmetry, in: British Machine Vision Conference (BMVC2008), The British Machine Vision Association and Society for Pattern Recognition, 2008, pp. 1115–1125.

[20] A. Torralba, A. Oliva, M. S. Castelhano, J. M. Henderson, Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search., Psychological review 113 (4) (2006) 766.

[21] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: Computer Vision, 2009 IEEE 12th international conference on, IEEE, 2009, pp. 2106–2113.

[22] A. Borji, Boosting bottom-up and top-down visual features for saliency estimation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 438–445.

[23] N. Bruce, J. Tsotsos, Saliency based on information maximization, Advances in neural information processing systems 18 (2006) 155.

[24] X. Shen, Y. Wu, A unified approach to salient object detection via low rank matrix recovery, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 853–860.

[25] D. A. Klein, S. Frintrop, Center-surround divergence of feature statistics for salient object detection, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2214–2219.

[26] W. Hou, X. Gao, D. Tao, X. Li, Visual saliency detection using information divergence, Pattern Recognition 46 (10) (2013) 2658–2669.

[27] X. Hou, L. Zhang, Dynamic visual attention: Searching for coding length increments, in: Advances in neural information processing systems, 2009, pp. 681–688.

[28] Z. Ren, S. Gao, L.-T. Chia, D. Rajan, Regularized feature reconstruction for spatio-temporal saliency detection, IEEE Transactions on Image Processing 22 (8) (2013) 3120–3132.

[29] G. Li, Y. Yu, Visual saliency detection based on multiscale deep cnn features, IEEE Transactions on Image Processing 25 (11) (2016) 5012–5024.

[30] H. Li, J. Chen, H. Lu, Z. Chi, Cnn for saliency detection with low-level feature integration, Neurocomputing 226 (2017) 212–220.

[31] C. Shen, M. Song, Q. Zhao, Learning high-level concepts by training a deep network on eye fixations, in: NIPS Deep Learning and Unsup Feat Learn Workshop, Vol. 2, 2012.

[32] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, Artificial Neural Networks and Machine Learning–ICANN 2011 (2011) 52–59.

[33] G. Kootstra, B. de Boer, L. R. Schomaker, Predicting eye fixations on complex visual stimuli using local symmetry, Cognitive computation 3 (1) (2011) 223–240.

[34] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3166–3173.

[35] B. W. Tatler, The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions, Journal of vision 7 (14) (2007) 4–4.

[36] R. J. Peters, A. Iyer, L. Itti, C. Koch, Components of bottom-up gaze allocation in natural images, Vision research 45 (18) (2005) 2397–2416.

[37] S. Goferman, L. Zelnik-Manor, A. Tal, Context-aware saliency detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (10) (2012) 1915–1926. doi:10.1109/TPAMI.2011.272.

[38] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in neural information processing systems, 2007, pp. 545–552.

[39] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Computer Vision and Pattern Recognition, 2014, pp. 2798–2805.

[40] R. B. Palm, Prediction as a candidate for learning deep hierarchical models of data, Technical University of Denmark.